1.Modified Grammar

- START -> PROGRAM
- PROGRAM -> DECL PROGRAM
- PROGRAM -> FUNCTION PROGRAM
- PROGRAM -> ''
- STMTS -> STMT STMTS
- STMTS -> STMT
- CODE_BLOCK -> STMT
- CODE_BLOCK -> SYM_P1 STMTS SYM_P2
- STMT -> DECL
- STMT -> ASSIGNMENT
- STMT -> CONDITIONAL
- STMT -> FOR
- STMT -> WHILE
- STMT -> DOWHILE
- STMT -> JUMP
- STMT -> SELECT
- STMT -> EXPR SYM_SEMICOLON
- DECL -> TYPE IDENTIFIER SYM_SEMICOLON
- ASSIGNMENT -> IDENTIFIER OP_ASSGN EXPR SYM_SEMICOLON
- ASSIGNMENT -> IDENTIFIER SYM_SB1 DIM SYM_SB2 OP_ASSGN EXPR SYM_SEMICOLON
- ASSIGNMENT -> IDENTIFIER SHORT_OP EXPR SYM_SEMICOLON
- ASSIGNMENT -> IDENTIFIER SYM_SB1 DIM SYM_SB2 SHORT_OP EXPR SYM_SEMICOLON
- CONDITIONAL -> KEYWD_IF SYM_B1 EXPR SYM_B2 CODE_BLOCK KEYWD_ELSE CODE_BLOCK
- FOR -> KEYWD_FOR IDENTIFIER KEYWD_IN SYM_B1 EXPR OP_COMMA EXPR SYM_B2 CODE_BLOCK

- WHILE -> KEYWD_WHILE SYM_B1 EXPR SYM_B2 CODE_BLOCK
- DOWHILE -> KEYWD_DO CODE_BLOCK KEYWD_WHILE SYM_B1 EXPR SYM_B2 SYM_SEMICOLON
- JUMP -> KEYWD_RETURN EXPR SYM_SEMICOLON
- JUMP -> KEYWD_BREAK SYM_SEMICOLON
- JUMP -> KEYWD_CONTINUE SYM_SEMICOLON
- SELECT -> KEYWD_SELECT SYM_B1 EXPR SYM_B2 SYM_P1 CASE_LIST SYM_P2
- CASE_LIST -> KEYWD_CASE EXPR OP_TERN1 STMTS CASE_LIST
- CASE_LIST -> KEYWD_DEFAULT OP_TERN1 STMTS
- CASE_LIST -> ''
- FUNCTION -> KEYWD_DEF IDENTIFIER SYM_B1 ARG_LIST SYM_B2 OP_FUNC TYPE SYM_P1 STMTS SYM_P2
- ARG_LIST -> TYPE IDENTIFIER OP_COMMA ARG_LIST
- ARG_LIST -> TYPE IDENTIFIER
- ARG_LIST -> ''
- FUNCTION_CALL -> IDENTIFIER SYM_B1 PARAM_LIST SYM_B2
- PARAM_LIST -> PARAMS
- PARAM_LIST -> ''
- PARAMS -> EXPR
- PARAMS -> EXPR OP_COMMA EXPR
- TYPE -> D_TYPE
- TYPE -> D_TYPE SYM_SB1 DIM SYM_SB2
- DIM -> EXPR OP_COMMA DIM
- DIM -> EXPR
- DIM -> ''
- D_TYPE -> KEYWD_INT
- D_TYPE -> KEYWD_FLOAT
- D_TYPE -> KEYWD_CHAR
- D_TYPE -> KEYWD_BOOL
- D_TYPE -> KEYWD_VOID
- D_TYPE -> KEYWD_STRING

- EXPR -> SYM_B1 EXPR BIN_OP EXPR SYM_B2
- EXPR -> PREFIX EXPR
- EXPR -> FUNCTION_CALL
- EXPR -> IDENTIFIER
- EXPR -> LITERAL
- EXPR -> SYM_P1 LIST SYM_P2
- LIST -> EXPR OP_COMMA LIST
- LIST -> EXPR
- LITERAL -> INT_LIT
- LITERAL -> FLOAT_LITERAL
- LITERAL -> CHAR_LIT
- LITERAL -> BOOL_LIT
- LITERAL -> STR_LIT
- BOOL_LIT -> KEYWD_TRUE
- BOOL_LIT -> KEYWD_FALSE
- BIN_OP -> ARTM_OP
- BIN_OP -> REL_OP
- BIN_OP -> LOGCL_OP
- BIN_OP -> BIT_OP
- ARTM_OP -> OP_AR_add
- ARTM_OP -> OP_AR_sub
- ARTM_OP -> OP_AR_mul
- ARTM_OP -> OP_AR_div
- ARTM_OP -> OP_AR_mod
- REL_OP -> OP_REL_gt
- REL_OP -> OP_REL_ge
- REL_OP -> OP_REL_lt
- REL_OP -> OP_REL_le
- REL_OP -> OP_REL_eq
- REL_OP -> OP_REL_neq
- LOGCL_OP -> OP_LOG_and

- LOGCL_OP -> OP_LOG_or
- LOGCL_OP -> OP_LOG_not
- BIT_OP -> OP_BIT_and
- BIT_OP -> OP_BIT_or
- BIT_OP -> OP_BIT_ls
- BIT_OP -> OP_BIT_rs
- BIT_OP -> OP_BIT_xor
- SHORT_OP -> OP_ASSGN1
- SHORT_OP -> OP_ASSGN2
- SHORT_OP -> OP_ASSGN3
- SHORT_OP -> OP_ASSGN4
- SHORT_OP -> OP_ASSGN5
- SHORT_OP -> OP_ASSGN6
- SHORT_OP -> OP_ASSGN7
- SHORT_OP -> OP_ASSGN8
- PREFIX -> OP_INCR
- PREFIX -> OP_DECR
- PREFIX -> OP_BIT4
- PREFIX -> OP_AR_add
- PREFIX-> OP_AR_sub

2.Non Terminals

1. START
2. PROGRAM
3. STMTS
4. CODE_BLOCK
5. STMT
6. DECL
7. ASSIGNMENT
8. CONDITIONAL
9. FOR

10. WHILE
11. DOWHILE
12. JUMP
13. SELECT
14. CASE_LIST
15. FUNCTION
16. ARG_LIST
17. FUNCTION_CALL
18. PARAM_LIST
19. PARAMS
20. TYPE
21. DIM
22. D_TYPE
23. EXPR
24. LIST
25. LITERAL
26. BOOL_LIT
27. BIN_OP
28. ARTM_OP
29. REL_OP
30. LOGCL_OP
31. BIT_OP
32. SHORT_OP
33. PREFIX

3.Terminals

1. SYM_P1
2. SYM_P2
3. SYM_SEMICOLON
4. IDENTIFIER
5. OP_ASSGN
6. SYM_SB1
7. SYM_SB2
8. KEYWD_IF

9.SYM_B1
10. SYM_B2
11. KEYWD_ELSE
12. KEYWD_FOR
13. KEYWD_IN
14. OP_COMMA
15. KEYWD_WHILE
16. KEYWD_DO
17. KEYWD_RETURN
18. KEYWD_BREAK
19. KEYWD_CONTINUE
20. KEYWD_SELECT
21. KEYWD_CASE
22. OP_TERN1
23. KEYWD_DEFAULT
24. KEYWD_DEF
25. OP_FUNC
26. KEYWD_INT
27. KEYWD_FLOAT
28. KEYWD_CHAR
29. KEYWD_BOOL
30. KEYWD_VOID
31. KEYWD_STRING
32. INT_LIT
33. FLOAT_LITERAL
34. CHAR_LIT
35. STR_LIT
36. KEYWD_TRUE
37. KEYWD_FALSE
38. OP_AR_add
39. OP_AR_sub
40. OP_AR_mul
41. OP_AR_div
42. OP_AR_mod

43. OP_REL_gt
44. OP_REL_ge
45. OP_REL_lt
46. OP_REL_le
47. OP_REL_eq
48. OP_REL_neq
49. OP_LOG_and
50. OP_LOG_or
51. OP_LOG_not
52. OP_BIT_and
53. OP_BIT_or
54. OP_BIT_ls
55. OP_BIT_rs
56. OP_BIT_xor
57. OP_ASSGN1
58. OP_ASSGN2
59. OP_ASSGN3
60. OP_ASSGN4
61. OP_ASSGN5
62. OP_ASSGN6
63. OP_ASSGN7
64. OP_ASSGN8
65. OP_INCR
66. OP_DECR
67. OP_BIT4

4.SDD

```
NTs have attributes code, temp
Ts have attributes val

  1.START -> PROGRAM
      a.
  2.PROGRAM -> DECL PROGRAM
      a.
  3.PROGRAM -> FUNCTION PROGRAM
      a.
  4.PROGRAM -> ''
      a.
  5.STMTS -> STMT STMTS
      a.Stmts.code = stmt.code||stmts.code
  6.STMTS -> STMT
      a.Stmts.code = stmt.code
  7.CODE_BLOCK -> STMT
      a.CODE_BLOCK.code = stmt.code
  8.CODE_BLOCK -> SYM_P1 STMTS SYM_P2
      a.CODE_BLOCK.code = stmts.code
  9.STMT -> DECL
      a.
  10. STMT -> ASSIGNMENT
      a.Stmt.code = ASSIGNMENT.code
  11. STMT -> CONDITIONAL
      a.Stmt.code = CONDITIONAL.code
  12. STMT -> FOR
      a.
  13. STMT -> WHILE
      a.
  14. STMT -> DOWHILE
      a.
  15. STMT -> JUMP
      a.
```

```
16. STMT -> SELECT
    a.
17. STMT -> EXPR SYM_SEMICOLON
    a.Stmt.code = expr.code
18. DECL -> TYPE IDENTIFIER SYM_SEMICOLON
    a.
19. ASSIGNMENT -> IDENTIFIER OP_ASSGN EXPR SYM_SEMICOLON
    a.ASSIGNMENT.code = EXPR.code||(IDENTIFIER.val =
      EXPR.temp)
20. ASSIGNMENT -> IDENTIFIER SYM_SB1 DIM SYM_SB2 OP_ASSGN
  EXPR SYM_SEMICOLON
    a.
21. ASSIGNMENT -> IDENTIFIER SHORT_OP EXPR SYM_SEMICOLON
    a.ASSIGNMENT.code = EXPR.code||(IDENTIFIER.val
      SHORT_op.temp EXPR.temp)
22. ASSIGNMENT -> IDENTIFIER SYM_SB1 DIM SYM_SB2 SHORT_OP
  EXPR SYM_SEMICOLON
    a.
23. CONDITIONAL -> KEYWD_IF SYM_B1 EXPR SYM_B2 CODE_BLOCK
  KEYWD_ELSE CODE_BLOCK
    a.CONDITIONAL.code = EXPR.code || if EXPR.temp = 0 goto
      elseLabel CODE_BLOCK1.code goto NextLabel elseLabel
      CODE_BLOCK2.code NextLabel
24. FOR -> KEYWD_FOR IDENTIFIER KEYWD_IN SYM_B1 EXPR OP_COMMA
  EXPR SYM_B2 CODE_BLOCK
    a.
25. WHILE -> KEYWD_WHILE SYM_B1 EXPR SYM_B2 CODE_BLOCK
    a.
26. DOWHILE -> KEYWD_DO CODE_BLOCK KEYWD_WHILE SYM_B1 EXPR
  SYM_B2 SYM_SEMICOLON
    a.
27. JUMP -> KEYWD_RETURN EXPR SYM_SEMICOLON
    a.
28. JUMP -> KEYWD_BREAK SYM_SEMICOLON
    a.
29. JUMP -> KEYWD_CONTINUE SYM_SEMICOLON
```

    a.

30. SELECT -> KEYWD_SELECT SYM_B1 EXPR SYM_B2 SYM_P1 CASE_LIST SYM_P2
    a.
31. CASE_LIST -> KEYWD_CASE EXPR OP_TERN1 STMTS CASE_LIST
    a.
32. CASE_LIST -> KEYWD_DEFAULT OP_TERN1 STMTS
    a.
33. CASE_LIST -> ''
    a.
34. FUNCTION -> KEYWD_DEF IDENTIFIER SYM_B1 ARG_LIST SYM_B2 OP_FUNC TYPE SYM_P1 STMTS SYM_P2
    a. FUNCTION.code=STMTS.code
35. ARG_LIST -> TYPE IDENTIFIER OP_COMMA ARG_LIST
    a.
36. ARG_LIST -> TYPE IDENTIFIER
    a.
37. ARG_LIST -> ''
    a.
38. FUNCTION_CALL -> IDENTIFIER SYM_B1 PARAM_LIST SYM_B2
    a.
39. PARAM_LIST -> PARAMS
    a.
40. PARAM_LIST -> ''
    a.
41. PARAMS -> EXPR
    a.
42. PARAMS -> EXPR OP_COMMA EXPR
    a.
43. TYPE -> D_TYPE
    a.
44. TYPE -> D_TYPE SYM_SB1 DIM SYM_SB2
    a.
45. DIM -> EXPR OP_COMMA DIM
    a.
46. DIM -> EXPR

```
    a.
47. DIM -> ''
    a.
48. D_TYPE -> KEYWD_INT
    a.
49. D_TYPE -> KEYWD_FLOAT
    a.
50. D_TYPE -> KEYWD_CHAR
    a.
51. D_TYPE -> KEYWD_BOOL
    a.
52. D_TYPE -> KEYWD_VOID
    a.
53. D_TYPE -> KEYWD_STRING
    a.
54. EXPR -> SYM_B1 EXPR BIN_OP EXPR SYM_B2
    a.EXPR.temp = newTemp();
    b.EXPR.code = EXPR.code || EXPR.code || newCode
    c.If BIN_OP!=relops newCode:EXPR.temp = EXPR.temp
      BIN_OP.temp EXPR.temp
    d.Else newCode:
       i.   L1 = newLabel()
      ii.   L2 = newLabel()
     iii.   if EXPR.temp BIN_OP.temp EXPR.temp goto L1
            EXPR.temp=0 goto L2 L1: EXPR.temp=1 L2:
55. EXPR -> PREFIX EXPR
    a.EXPR.temp = newTemp()
    b.EXPR.code = EXPR.code||EXPR.temp = PREFIX.temp
      EXPR.temp
56. EXPR -> FUNCTION_CALL
    a.
57. EXPR -> IDENTIFIER
    a.EXPR.temp = IDENTIFIER.val
58. EXPR -> LITERAL
    a.EXPR.temp = LITERAL.temp
59. EXPR -> SYM_P1 LIST SYM_P2
```

```
    a.
60. LIST -> EXPR OP_COMMA LIST
    a.
61. LIST -> EXPR
    a.
62. LITERAL -> INT_LIT
    a.LITERAL.temp = INT_LIT.val
63. LITERAL -> FLOAT_LITERAL
    a.LITERAL.temp = FLOAT_LIT.val
64. LITERAL -> CHAR_LIT
    a.LITERAL.temp = CHAR_LIT.val
65. LITERAL -> BOOL_LIT
    a.LITERAL.temp = BOOL_LIT.temp
66. LITERAL -> STR_LIT
    a.
67. BOOL_LIT -> KEYWD_TRUE
    a.BOOL_LIT.temp = KEYWD_TRUE.val
68. BOOL_LIT -> KEYWD_FALSE
    a.BOOL_LIT.temp = KEYWD_FALSE.val
69. BIN_OP -> ARTM_OP
    a.BIN_OP.temp = ARTM_OP.temp
70. BIN_OP -> REL_OP
    a.BIN_OP.temp = REL_OP.temp
71. BIN_OP -> LOGCL_OP
    a.BIN_OP.temp = LOGCL_OP.temp’
72. BIN_OP -> BIT_OP
    a.BIN_OP.temp = BIT_OP.temp
73. ARTM_OP -> OP_AR_add
    a.ARTM_OP.temp = OP_AR_add.val
74. ARTM_OP -> OP_AR_sub
    a.ARTM_OP.temp = OP_AR_sub.val
75. ARTM_OP -> OP_AR_mul
    a.ARTM_OP.temp = OP_AR_mul.val
76. ARTM_OP -> OP_AR_div
    a.ARTM_OP.temp = OP_AR_div.val
77. ARTM_OP -> OP_AR_mod
```

```
    a.ARTM_OP.temp = OP_AR_mod.val
78. REL_OP -> OP_REL_gt
    a.REL_OP.temp = OP_REL_gt.val
79. REL_OP -> OP_REL_ge
    a.REL_OP.temp = OP_REL_ge.val
80. REL_OP -> OP_REL_lt
    a.REL_OP.temp = OP_REL_lt.val
81. REL_OP -> OP_REL_le
    a.REL_OP.temp = OP_REL_le.val
82. REL_OP -> OP_REL_eq
    a.REL_OP.temp = OP_REL_eq.val
83. REL_OP -> OP_REL_neq
    a.REL_OP.temp = OP_REL_neq.val
84. LOGCL_OP -> OP_LOG_and
    a.LOGCL_OP.temp = OP_LOG_and.val
85. LOGCL_OP -> OP_LOG_or
    a.LOGCL_OP.temp = OP_LOG_or.val
86. LOGCL_OP -> OP_LOG_not
    a.LOGCL_OP.temp = OP_LOG_not.val
87. BIT_OP -> OP_BIT_and
    a.BIT_OP.temp = OP_BIT_and.val
88. BIT_OP -> OP_BIT_or
    a.BIT_OP.temp = OP_BIT_or.val
89. BIT_OP -> OP_BIT_ls
    a.BIT_OP.temp = OP_BIT_ls.val
90. BIT_OP -> OP_BIT_rs
    a.BIT_OP.temp = OP_BIT_rs.val
91. BIT_OP -> OP_BIT_xor
    a.BIT_OP.temp = OP_BIT_xor.val
92. SHORT_OP -> OP_ASSGN1
    a.SHORT_OP.temp = OP_ASSGN1.val
93. SHORT_OP -> OP_ASSGN2
    a.SHORT_OP.temp = OP_ASSGN2.val
94. SHORT_OP -> OP_ASSGN3
    a.SHORT_OP.temp = OP_ASSGN3.val
95. SHORT_OP -> OP_ASSGN4
```

```
    a.SHORT_OP.temp = OP_ASSGN4.val
96. SHORT_OP -> OP_ASSGN5
    a.SHORT_OP.temp = OP_ASSGN5.val
97. SHORT_OP -> OP_ASSGN6
    a.SHORT_OP.temp = OP_ASSGN6.val
98. SHORT_OP -> OP_ASSGN7
    a.SHORT_OP.temp = OP_ASSGN7.val
99. SHORT_OP -> OP_ASSGN8
    a.SHORT_OP.temp = OP_ASSGN8.val
100.PREFIX -> OP_INCR
    a.PREFIX.temp = OP_INCR.val
101.PREFIX -> OP_DECR
    a.PREFIX.temp = OP_DECR.val
102.PREFIX -> OP_BIT4
    a.PREFIX.temp = OP_BIT4.val
103.PREFIX -> OP_AR_add
    a.PREFIX.temp = OP_AR_add.val
104.PREFIX-> OP_AR_sub
    a.PREFIX.temp = OP_AR_sub.val
```